

RC 4000 Programudvikling

PBH/30.10.1967

REFERAT AF MONITOR MØDE PÅ MARINA, 28 - 29 OKTOBER 1967

Projektgruppen, der arbejder med RC 4000 monitoren, har i nogen tid følt at Grenaa oplagget (karakteriseret ved en fast programopdeling i sekund-, minut-, og timekørsler) ville føre til et stift system med mange restriktioner og ad hoc mekanismer.

Formålet med Marina mødet var at skitsere en mere generel ramme for parallel kørsel på RC 4000. Hvis dette ikke var lykkedes på mødet, var deltagerne indstillet på kun at implementere en meget begrænset form for parallel kørsel på RC 4000.

Vi nåede imidlertid frem til en generel systemramme, som vi mener er et realistisk arbejdsgrundlag for en RC 4000 monitor. Hovedideen heri er en styring af de ydre enheder, der simplificerer dumpning og genstart af programmer.

Per Brinch Hansen Jørn Jensen Peter Kraft Søren Lauesen

Parallel Kørsel

I dette afsnit defineres målsætningen for systemrammen.

Parallel kørsel af flere sekventielle programmer på en datamaskine sker ved en tidsdeling af centralenheden og det ydre udstyr blandt programmerne. Hermed tilstræbes to ting: at udnytte det samlede udstyr bedst muligt, samt at tillade adgang til maskinen med kort varsel.

For at opnå dette vil vi først og fremmest tillade monitorstyret tidsdeling blandt flere programmer, der befinder sig samtidigt i hovedlageret. Det betyder at brugerne får øjeblikkelig adgang til maskinen, sålænge der er lagerplads og ledige ydre enheder. Et programs ventetid på ydre enheder vil her kunne udnyttes som procestid for andre programmer.

Vi ønsker ikke i oplægget at lægge nogen begrænsninger på de enkelte programmer (f.eks. ved at klassificere dem efter størrelse, køretid, krav om ydre enheder, osv.). Det tages derfor for givet, at visse programmer vil lægge beslag på størstedelen af hovedlageret og de ydre enheder.

Sådanne store programmer må ikke kunne blokere anlægget for andre brugere. Det skal derfor være muligt for operatøren at stoppe ethvert program midlertidigt på en sådan måde, at det senere kan genstartes. Når et program er stoppet, skal både dets hovedlagerplads og ydre enheder være tilgængelige for andre programmer.

I det følgende skitseres en systemramme, der opfylder disse krav.

Ydre Enheder

Når et program genstartes, skal filer på ydre enheder kunne reetableres i den position de havde, da programmet blev stoppet. En sådan reetablering er imidlertid også nødvendig under en programkørsel, hvis operatøren foretager utilsigtede indgreb i de ydre enheder, eller hvis en fil flyttes fra en enhed til en anden (fordi den første enhed er brudt sammen).

Det er derfor ønskeligt principielt at betragte enhver datatransport som en genstart i den forstand, at det skal kontrolleres, at den ønskede fil stadig er monteret i den ønskede position.

For at opnå dette oprettes i monitoren en enbedstabel. Hver indgang i denne tabel beskriver en ydre enhed ved dens nummer og type (magnetbånd, linieskriver, kodebåndslaser, etc). Endvidere beskrives den fil, der

øjeblikket er monteret på enheden, ved et filnavn og en filposition (segment no, blok no, tegn no, etc). Under en datatransport indeholder tabellen desuden navnet på det program, der har startet transporten, samt diverse oplysninger om transportens forløb (paritetsfejl o.lign.):

<enhedstype> <enhedsnummer> <fejlhyppighed>
<filnavn> <filposition>
<programnavn> <transport status>

Systemet må kunne registrere, når operatøren piler ved en ydre enhed (f.eks. ved at der sættes en statusbit i enheden). Når det sker vil beskrivelsen i enhedstabellen øjeblikkelig blive ændret til: ukendt fil.

Når operatøren har monteret en fil på en enhed kan han fra et konsol placere filens navn i enhedstabellen. For magnetbånd kan filnavnet være et kasettenummer, der kontrolleres af monitoren ved sammenligning med båndetiketten. For kodebånd og linieskriverformularer akcepteres filnavnet antageligt uden kontrol.

Når et program refererer til en ydre enhed overfører det følgende parametre til monitoren:

<filnavn> <filposition> <operation> <bufferadresse>

Betingelsen for at starte operationen er nu den, at filen eksisterer i den ønskede position. Dette afgøres ved hjælp af enhedstabellen. Hvis filen er ukendt hoppes retur til programmet, der så må bede operatøren om at montere den på en ledig enhed.

Datatransporter sker direkte til lokale bufferarealer indenfor programmerne.

Baggrundslager

Baggrundslageret tænkes opdelt i navngivne områder, der er beskrevet i et katalog (der selv befinder sig på baggrundslageret).

Et navngivet område består af et antal konsekutive datablokke.

Navngivne områder, der bruges af kørende programmer, er beskrevet i enhedstabellen sammen med de egentlige ydre enheder, således at deres eksistens kan kontrolleres før hver datatransport.

Når et navngivet område nedlægges eller flyttes på baggrundslageret, fjernes samtidigt en eventuel beskrivelse af området fra enhedstabellen. Et program, der herefter refererer til området, vil få at vide at det er ukendt. Hvis det blot er flyttet, kan beskrivelsen i enhedstabellen reetableres ved et katalogopslag.

Konsoller

Programmerne kommunikerer under kørslen med et antal brugere og en hovedoperatør ved hjælp af en række konsoller. Hovedoperatøren er den, der tager stilling til de programkrav, der kun kan opfyldes ved at gribe ind i andre programmer (f.eks. beslutningen om at dumpe et kørende program for at starte et andet).

Konsolkonversationer afvikles parallelt med kørslen af brugerprogrammer.

En bruger kan gå hen til et vilkårligt konsol og navngive det, på samme måde som han kan navngive andre ydre enheder. Der er dog en praktisk forskel mellem konsoller og andre ydre enheder, idet det er en normal situation at flere programmer på en gang henvender sig til samme konsol (f.eks. ved meddelelser til hovedoperatøren, samt naturligvis i installationer, hvor der kun findes et konsol). Der må derfor være en kø af meddelelser for hvert konsol.

Et program kan henvende sig til et navngivet konsol og få udskrevet en meddelelse. Programmet kan endvidere specificere, om meddelelsen kræver et svar fra operatøren. I så fald må operatører kunne vælge om han vil svare med det samme eller om han vil udskyde svaret på en kompliceret forespørgsel (for ikke at blokere konsollet for andre programmer). Dette kræver at han, når svaret foreligger, identificerer det program svaret er rettet til. En løsning på dette problem er skitseret af E.W. Dijkstra (Cooperating Sequential Processes, side 48 -73).

Lagerallokering

RC 4000s adresseringstruktur er uegnet til en generel segmentering af programmer i hovedlageret. Fra monitorens synspunkt består et program derfor simpelthen af et enkelt konsekutivt område i hovedlageret.

Et program kan placeres hvorsomhelst i lageret, når det startes forfra.

Et program, der stoppes midlertidigt og dumpes på baggrundslageret, skal ved genstart indlases til samme plads i hovedlageret.

Stop og Genstart af Programmer

Et program kan stoppes midlertidigt ved at monitoren venter på samtlige af programmets igangværende ydre enheder og derefter dumper dets lagerbillede og arbejdsregistre på baggrundslageret. Dette indebærer ikke nogen ændring af enhedstabellen med mindre operatøren ændrer filnavne for at kunne bruge enhederne til et andet program.

Ved senere indlæsning af programmet til samme lagerareal og efter reetablering af arbejdsregistre, kan det umiddelbart genstartes. Hvis nogle af programmets filer imellemtiden er blevet fjernet informeres programmet automatisk herom, når det forsøger at referere til dem.

Arbejdsplan

Den detaljerede udarbejdning af monitoren er planlagt at ske i følgende faser:

1. Konsolkonversationer
2. Andre ydre enheder
3. Navngivne områder på baggrundslageret
4. Beskrivelse af programtilstande og overgange mellem disse
5. Beskyttelsesproblemer